

```

+-----+
| CCC   000   N   N   QQQ   U   U   EEEEE   SSSS   TTTT |
| C   C   O   O   NN   N   Q   Q   U   U   E       S       T   |
| C       O   O   N   N   N   Q   Q   U   U   EEE       SSS   T   |
| C   C   O   O   N   NN   Q   Q   U   U   E           S       T   |
| CCC   000   N   N   QQ   Q   UUU   EEEEE   SSSS       T   |
+-----+

```

Server Guide

So you want to run a Conquest Server? Then this is the file to read.

It gives you an overview of how to setup and manage one, as well as other things to keep in mind.

If all you want to do is use conquest to play on someone else's server, then you do not need to be reading this document. :)

In the text below, *prefix* refers to the installation prefix specified to configure when your conquest was built. This defaults to /opt.

OVERVIEW

With version 8.0 of Conquest, a great deal of work has gone into making Conquest a true client/server game playable over the Internet (or just your local LAN).

Conquest has two main parts, the server component (conquestd) and the client: conquest.

Version 8.0.1c or better includes many server performance improvements, including UDP support for motion data.

To run a server, you must run the conquestd program.

Clients (including ones on your local machine) then connect to your server and the game is played.

Running a server

conquestd is used to provide a Conquest game. Even if all you want to do is play by yourself on your own machine, you will need to run conquestd in order to do so.

conquestd supports a variety of options outlined below:

```

$ conquestd -?
Usage: conquestd [ -d ] [ -l ] [ -p port ] [ -u user ]
                [ -m ] [ -M metaserver ] [ -N myname ]

  -d                daemon mode
  -G subdir         specify alternate game subdirectory
  -l                listen for local connections only
  -p port           specify port to listen on
                   default is 1701
  -m                notify the metaserver (conquest.radscan.com)
  -M metaserver     specify an alternate metaserver to contact
  -N myname         explicitly specify server name 'myname' to metaserver
  -u user           run as user 'user'.

```

Insecure start

In it's simplest (and insecure) form, you can simply run:

```
$ conquestd -d
```

This will start conquestd, which will fork itself into the background and run as your user id.

It will listen on the default port (1701) for client connections, and spawn a new conquestd for each client connection. It will be available to anyone who can connect to TCP port 1701 on your machine.

I would only run a server this way if no one else from the Internet will be able to connect (if you are behind a good firewall for example). This is insecure since conquestd will be running as your uid. Read further for a more secure way to run the server.

Secure start

For the secure case, I would start conquestd as root, and have it run as user 'nobody' by passing the '-u user' option on the command line. Of course you can create and use some other non-privileged, non-interactive user for this task as well.

Something like the following is recommended:

```
# conquestd -d -u nobody
```

This will have conquestd `setuid()` permanently to user 'nobody' before beginning operations. On most unix systems, user 'nobody' has no home directory, password, or privileges, and therefore is a safe user id to run under.

You must be the root user in order to use the '-u' option.

DO NOT run conquestd as the root user itself. Although the code *should* be safe, you shouldn't take my word for it.

Notifying the metaserver

If you are going to run a server that will be available to the public via the Internet, you can pass the '-m' option to conquestd when you start it so it will notify the metaserver at conquest.radscan.com.

When running with the '-m' option, conquestd will contact the Conquest metaserver and announce your server, so that other people will know about it.

Every 2 minutes or so, it will update the metaserver with various particulars of your server (like how many players are currently connected, what flags you support, what your server name, version, and MOTD are, etc).

This way, other people can find out about your server and connect to it to play.

The list of currently active servers can be seen in the game by running your client without any options.

Additionally, you can also point your web browser to:

<http://radscan.com/cgi-bin/conqmeta.pl>

Please test and make sure that your server is actually reachable from the Internet before notifying the metaserver about it.

Firewall considerations

Most people that will run a server will also be using a firewall of some sort to protect their networks.

See the [README](#) file for the ports that conquest typically uses.

Inbound access

If you wish to allow internet access to your server, you will need to allow inbound TCP *and* UDP access to the game port (1701 by default).

Outbound access

If you restrict outbound internet connections, you might want to allow the following outgoing ports:

1700/tcp - if you want clients to be able to query the metaserver for active servers.

1700/udp - if you are running a server, and you want it to be able to announce it's availability to the metaserver at conquest.radsan.com.

Expiring users

By default, whenever a user logs into Conquest, an autoexpire is run to locate and 'resign' all inactive remote users.

A user is expired:

- i. if the user is a non OPER or non robot user - no expiration is ever done on these users.
- ii. the user hasn't entered the game in 'user_expiredays' days, which is set in the system-wide conquest.conf file.
- iii. the user isn't currently flying a ship.

If all of these conditions are met, the remote user is resigned from the game.

Disabling expiration of users

If you wish to disable expiration altogether, set 'user_expiredays' equal to 0 in the system-wide conquest.conf file, or via the Options menu in conqoper.

Server start examples

Here are a few examples for starting the server (the '#' prompt implies you are running these as root):

```
# conquestd -d -u nobody
```

Runs a server as user nobody.

```
# conquestd -d -u nobody -l
```

Runs a server as user nobody. Only local clients can connect.

```
# conquestd -d -u nobody -m
```

Runs a server as user nobody. Notifies the Conquest metaserver about your server.

```
# conquestd -d -u nobody -m -N conquest.radscan.com
```

Runs a server as user nobody, notifies the Conquest metaserver of the game, and tells the metaserver that the address 'conquest.radscan.com' is the real address of your server, as seen from the Internet.

Conqoper

Conqoper is a curses based utility program for controlling and modifying your game configuration.

Fire it up without options and you will be taken to the main menu. It should be pretty straightforward to use.

It also supports certain operations via options passed on the command line.

```
$ conqoper -?
conqoper: invalid option -- '?'
usage: conqoper [-C] [-D] [-E] [-I <what>]
  -C                rebuild systemwide conquest.conf file
  -D                disable the game
  -E                enable the game
  -G                specify alternate game subdirectory
  -I <what>         Initialize <what>, where <what> is:
    e - everything
    g - game
    l - lockwords
    m - messages
    p - planets
    r - robots
    s - ships
    u - universe
    z - zero common block
```

Permissions needed to run conqoper

To run conqoper you must either be the root user, or your username must be a member of the conquest group ("conquest", by default).

Initializing the Universe

When running a server, one of the first things you will need to do is initialize the Universe using conqoper. You can do this on the command line:

```
conqoper -Ie
```

Note, this wipes out any information in your Universe and resets it from scratch, so be careful. Also, don't do this while people are playing :)

You will also need to re-initialize the universe any time you modify any of the values in the "global {}" section of the *conqinitrc* file.

That file specifies some limits on the universe (number of ships supported, planets supported, etc) as well as definitions for each of the planets in your universe.

Configuration files

There are a few configuration files in *prefix/etc/conquest/* worth noting.

conqinitrc

For a server, the most important file is the *prefix/etc/conquest/conqinitrc* file (where *prefix* is whatever prefix you installed conquest into).

The *conqinitrc* file specifies the configuration of your universe - how many ships can fly at the same time, how many users, planets, etc are supported.

In addition, all planet definitions reside in this file. It is here, you can modify your Universe from the default shipped with the game.

Look through this file to see it's format. Note that modifying this file may require you to re-init the universe via *conqoper* from scratch.

texturesrc and soundrc

These configuration files are only needed by the client.

conquest.conf

This file is a simple configuration file used to set various options for your server. It will be created/located in *prefix/etc/conquest/conquest.conf*.

You can edit this file directly, or use the (O)ptions menu option in the *conqoper* program. It is here where you enable and disable various flags and options for your game, set your server name, MOTD, and other information.

It is automatically created on a “make install”, if it does not already exist (via *conqoper -C*).

With Conquest version updates the file is updated automatically, preserving previous settings.

Preserving local modifications to configuration files

Doing a “make install” or updating to a newer package will overwrite these files, so make *.local* copies of these files if you want to preserve your modifications.

For example, if you are running a server and want to make a customized game by making changes to the *conqinitrc* file:

```
cd *prefix*/etc/conquest
cp conqinitrc conqinitrc.local
vi conqinitrc.local
...
```

NOTE: As mentioned above, changing any of the items in the *global {}* section in *conqinitrc* will require that you re-init the universe from *conqoper* before you will be able to start a server. The server will warn you about this if you try to start it, with a “Common Block Mismatch” error. This is your cue to re-init the universe :)

Handling multiple games

Starting with version 9, a *-G* option has been added to the server components (*conquestd*, *conqoper*, *conqdriv*, etc). This allows a single conquest installation to support multiple games. Previously, if you wanted to serve multiple games you would need to compile a version of conquest for each one using a different *-prefix* option to *configure*.

The main files a server needs to know about a game are located in *prefix/etc/conquest/*, and *prefix/var/conquest/*, so we will create new sub directories there to support additional games.

It's important that the new directories have the correct ownership and permissions, so that the server components can access them, and in some cases write into them.

So, as an example, if you wanted to serve a second game called *testgame*, you would do something like the following, as the root user, to set it up initially (assuming *prefix* is */opt*):

```
cd /opt/etc/conquest
mkdir testgame
chgrp conquest testgame
chmod 775 testgame
cp conqinitrc testgame/
```

Then edit *testgame/conqinitrc* as desired for your new game. Next, we need to create a similar subdirectory in *prefix/var/conquest* to hold the new universe and it's log.

```
cd /opt/var/conquest
mkdir testgame
chgrp conquest testgame
chmod 775 testgame
```

You could then initialize it with:

```
/opt/bin/conqoper -G testgame -C -Ie
```

This command would create a new */opt/etc/conquest/testgame/conquest.conf* file (if it did not already exist), and then initialize a new universe in */opt/var/conquest/testgame/conquest.cb*.

You could then start a server to serve this new game with:

```
/opt/bin/conquestd -G testgame ...other server options...
```

Run *conqoper* interactively on 'testgame'

```
/opt/bin/conqoper -G testgame
```

All of the server components support the *-G* option. It is irrelevant for clients, so the option is not supported there.

CQI Parser

This is the parser responsible to decoding these various configuration files (except for *conquest.conf*, which is handled differently).

For a full description of their format, allowed values, etc, see the [CQI Parser Documentation](#).